

Vorlesung “Objektorientierte Programmierung” — Beispiel für 1. Programmierertest (Aufgabe A) —

Hinweise/Regeln:

Vermeiden Sie bitte jedes Verhalten, was als Täuschungsversuch (miss-)verstanden werden könnte. Wir müssten Sie sonst disqualifizieren, d.h. Sie hätten den Programmierertest nicht bestanden. Sie werden sowohl durch das Aufsichtspersonal im Rechnerpool überwacht, als auch auf den Rechnern per Programm/Aufzeichnung, sowie über das Netz.

- Bearbeitungsdauer: 70 Minuten (plus eventuell leichte Verlängerung).
- Es gibt nur “bestanden” oder “nicht bestanden”, keine Punkte für partiell korrekte Lösungen. Eventuell werden am Ende einige Minuten Bearbeitungszeit angehängt, wenn mehrere Teilnehmer knapp vor einer fertigen Lösung sind — aber nicht funktionierende Lösungen werden nicht akzeptiert, selbst wenn nur eine Kleinigkeit fehlt.
- Sie dürfen bis zu 3 Blätter “Spickzettel” / “Quick Reference” verwenden, sowie ein Buch (nicht zu groß, es muß noch auf den Tisch passen ohne den Nachbarn zu stören — ein Aktenordner wäre nicht akzeptabel). Sie dürfen außerdem ein Blatt leeres Papier und einen Stift verwenden, um sich Notizen zu machen.
- Mobiltelefone bitte ausschalten (oder mit der Aufsicht besprechen).
- Sie dürfen ein eigenes Notebook verwenden, wenn Sie es bisher in den Übungen benutzt haben. WLAN und sonstige Netzverbindungen müssen ausgeschaltet sein.
- Es ist nicht erlaubt, einen Web-Browser zu verwenden, ein EMail-Programm, oder sonstige Netzwerk-Zugriffe. Automatische Netzwerk-Zugriffe Ihrer Entwicklungsumgebung sind in Ordnung, aber ggf. geöffnete Webseiten schließen Sie bitte sofort.
- Die automatisch eingeblendeten Methoden-Signaturen und andere Hinweise gehören zur Entwicklungsumgebung, aber ansonsten verwenden Sie die Hilfe bitte nicht (insbesondere keine Suchfunktionen).
- Sie dürfen nicht versuchen, Dateien außerhalb Ihres Homeverzeichnisses abzulegen, oder auf Dateien außerhalb Ihres Homeverzeichnisses zuzugreifen (außer Dateien, die der Compiler bzw. die Entwicklungsumgebung verwendet).
- Auch auf vorhandene Dateien innerhalb Ihres Homeverzeichnisses dürfen Sie nicht zugreifen (außer Voreinstellungen etc. durch die Entwicklungsumgebung). Sie müssen daher das Programm vollständig neu eintippen, und nicht durch Modifikation von eventuell in Ihrem Homeverzeichnis bereits vorhandenen Dateien erstellen.
- Auch vorhandene Text-Dokumente (z.B. PDF) dürfen Sie nicht anzeigen.
- Die Homeverzeichnisse werden für Zugriffe von außen gesperrt. Falls Sie spezielle Zugriffsrechte gesetzt hatten, müssen Sie diese nach dem Test selbst wieder herstellen.

- Tauschen Sie keinesfalls irgendwelche Dinge mit den Nachbarn aus. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Sie müssen Mindestanforderungen an den Programmierstil erfüllen, z.B. entsprechend der Programmstruktur einrücken.
- Fragen Sie, wenn Ihnen die Aufgabe nicht klar ist! Wenn Sie an einer unverständlichen Fehlermeldung länger festhängen, können Sie probieren, zu fragen. Wir wollen aber nicht zu viele Tipps geben.

Aufgabe (Variante A)

Schreiben Sie ein Programm, mit welchem mehrere LKWs beladen werden können. Jeder LKW hat ein maximales Gewicht, das Ihrem Programm als Konsolenparameter übergeben wird. Die Anzahl der Konsolenparameter entspricht somit auch der Anzahl der LKWs.

Ihr Programm soll beim Beladen eines LKW-Anhängers mit verschiedenen schweren Containern darauf achten, dass das Gesamtgewicht nicht überschritten wird. Jeder LKW kann mit Containern unterschiedlichen Gewichts beladen werden. Wird mit der Beladung eines LKWs begonnen, soll dies auch ausgegeben werden. Ist der erste Konsolenparameter gleich 5000, so bedeutet dies, dass unser erster LKW ein Maximalgewicht von 5000 kg hat. Die geforderte Ausgabe würde somit wie folgt aussehen:

```
Beladung des 1. LKWs (max.: 5000 kg).
```

Nun wird in einer Schleife jeweils das Gewicht eines Containers, welcher auf den Hänger geladen werden soll, eingelesen. Für jeden Container soll der Text

```
Gewicht des naechsten Containers (in kg) eingeben:
```

ausgegeben werden (am Ende mit einem Leerzeichen, aber ohne Zeilenumbruch). Die Eingabe soll als ganze Zahl erfolgen (volle kg). Wenn der eingegebene Zahlwert $<$ oder $= 0$ ist, wird folgende Fehlermeldung ausgegeben:

```
Ein Container hat immer ein Gewicht groesser 0!
```

(mit einem Zeilenumbruch am Ende). Das Programm soll sich dann beenden. Sie können davon ausgehen, dass eine syntaktisch korrekte ganze Zahl eingegeben wurde. Prüfen Sie anschließend, ob mit diesem Container das Gesamtgewicht der Ladung überschritten wird. Falls nicht, so wird der Container aufgeladen. Wird das Gesamtgewicht der Ladung überschritten, so sollen folgende Meldungen ausgegeben werden:

```
Mit diesem Container wird das maximale Gesamtgewicht ueberschritten!  
Maximales Gewicht des naechsten Containers: 123 kg
```

Beide Ausgaben sollen jeweils mit einem Zeilenumbruch enden. Die Schleife wird beendet, wenn der Hänger voll ist. D.h., sein Maximalgewicht erreicht ist. Geben Sie dann eine entsprechende Mitteilung aus. Wurde z. B. der erste LKW voll beladen, würde die Ausgabe so aussehen:

```
Der Haenger des 1. LKWs ist voll!
```

Sie dürfen den folgenden Rahmen für Ihr Hauptprogramm verwenden:

Listing 1: Hauptprogramm

```
1 import java.io.*;
2
3 public class LKWs
4 {
5     public static void main(String [] args) throws IOException
6     {
7         BufferedReader buffer = new BufferedReader(
8             new InputStreamReader(System.in));
9         //...
10    }
11 }
```

Hinweis: Das Einlesen einer Zahl von der Konsole können Sie mittels `int n = Integer.parseInt(buffer.readLine());` realisieren. Ebenso erhalten Sie mit `Integer.parseInt(args[i])` den Zahlwert des *i*-ten Kommandozeilen-Arguments.

Beispiele für Tests:

- Bei einer negativen Eingabe muss die vorgegebene Fehlermeldung ausgegeben werden.
- Ein Beispiel-Lauf für den Aufruf

```
java LKWs 5000 3000 4000
```

ist:

```
Beladung des 1. LKWs (max.: 5000 kg).
Gewicht des naechsten Containers (in kg) eingeben: 3000
Gewicht des naechsten Containers (in kg) eingeben: 2500
Mit diesem Container wird das maximale Gesamtgewicht ueberschritten!
Maximales Gewicht des naechsten Containers: 2000 kg
Gewicht des naechsten Containers (in kg) eingeben: 2000
Der Haenger des 1. LKWs ist voll!
Beladung des 2. LKWs (max.: 3000 kg).
Gewicht des naechsten Containers (in kg) eingeben: 3000
Der Haenger des 2. LKWs ist voll!
Beladung des 3. LKWs (max.: 4000 kg).
Gewicht des naechsten Containers (in kg) eingeben: 2500
Gewicht des naechsten Containers (in kg) eingeben: 500
Gewicht des naechsten Containers (in kg) eingeben: 800
Gewicht des naechsten Containers (in kg) eingeben: 500
Mit diesem Container wird das maximale Gesamtgewicht ueberschritten!
Maximales Gewicht des naechsten Containers: 200 kg
Gewicht des naechsten Containers (in kg) eingeben: 200
Der Haenger des 3. LKWs ist voll!
```